Computer Vision
Assignment 2

Optical Flow

Jacob Spieß: dieter.jacob.spiess@vub.be
December 2022

Abstract

In the following report an implementation of the Horn - Schunck method for calculating the optical flow is discussed. Hereby, first the optical flow is explained, afterwards it is described how the Horn - Schunck method can be used to estimate the optical flow, lastly example results from the implementation of the method are discussed.

Optical flow

When looking at a sequence of pictures e.g., in a video, it is of interest to capture the change that happens from one sequence to another. To realize that the motion of objects needs to be tracked across all frames, by estimating the current position and predicting the new position in the picture happening at *t+1*. In addition to the space coordinates x and y, when working with a sequence of pictures the change of intensity of a pixel over time t becomes relevant. Hence, the intensity *I* of a pixel is a function of *x, y,* and *t* (Horn and Schunck, 1981).

Now the optical flow is the pattern of motion of pixels between two consecutive pictures, whereby movement can occur naturally in the scene or through moving the camera. The optical flow describes a motion field where each pixel is assigned a separate 2D displacement vector, which estimates the direction and speed of the physical movement relative to the observer (Horn and Schunck, 1981).

The key assumption to solving the optical flow is the brightness constancy, which means that there are no significant changes in the colours of the object, as the lighting source is fixed. This in combination with assuming only small motions between two pictures allows one to find pixel correspondences by looking at nearby pixels of the same colour in the second picture. Using the brightness constraint allows to derive the following equation:

$$E_x u + E_y v + E_t = 0$$

whereby the partial derivatives are known, and the unknowns (*u, v*) lay on a straight line. Hence, there are infinite possible values for *u* and *v*, and they cannot be figured out without any additional information. This is also called the aperture problem, which refers to the fact that the motion of structures like bars or edges, cannot be determined unambiguously if not viewed holistically (Horn and Schunck, 1981).

Horn-Schunck method

The Horn - Schunck method introduces the smoothness constraint to solve the equation, by assuming that neighbours in proximity move in the same direction and therefore having similar optical flow vectors. The idea is to introduce a smoothness constraint that minimizes the difference from neighbours. The Horn - Schunck method brings both constraints together by enforcing brightness constancy as well as a smooth flow field. This leads to a cost function where the smoothness plus the weighted brightness constancy is computed and summed for each pixel of the image, while trying to minimize *u* and *v* (Horn and Schunck, 1981).

$$\min_{u,v} \sum_{i,j} \{E_{\text{smoothness}}(i,j) + \lambda E_{\text{brightness}}(i,j)\}$$

To simplify the equation the partial derivatives with respect to *u* and *v* are computed and set to zero to solve for u and v.

$$\sum_{ij} \left\{ \frac{1}{4} \left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2 \right] \right.$$
$$\left. + \lambda \left[E_x u_{ij} + E_y v_{ij} + E_t \right]^2 \right\}$$

Considering the intermediate quotations as linear systems and rearranging the terms, the following two update rules can be obtained:

$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{E_x \bar{u}_{kl} + E_y \bar{v}_{kl} + E_t}{\lambda^{-1} + E_x^2 + E_y^2} E_x$$

$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{E_x \bar{u}_{kl} + E_y \bar{v}_{kl} + E_t}{\lambda^{-1} + E_x^2 + E_y^2} E_x$$

Whereby $\bar{u}_{kl}$ is the old average and $\hat{u}_{kl}$ the new value (Horn and Schunck, 1981).

Implementation

To implement the optical flow with the Horn-Schunck method first the flow fields *u* and *v* need to be initialized with zero values. Second, the image and temporal gradients need to be computed using two consecutive images as the basis. Lastly, the flow field updates for each pixel are computed in a loop, where the number of iterations is a hyperparameter that needs to be tuned. This translates to the following pseudo code:

```
def horn_schunck():
    initialize u
    initialize v
    calculate x_grad, y_grad, t_grad

    itterate:
        update u
        update v
    return u, v
```

References

B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*, vol 17, pp 185–203, 1981.